Technical Brief ■

Encryption Characteristics of Two USB-based Personal Health Record Devices

ADAM WRIGHT, PHD, DEAN F. SITTIG, PHD

Abstract Personal health records (PHRs) hold great promise for empowering patients and increasing the accuracy and completeness of health information. We reviewed two small USB-based PHR devices that allow a patient to easily store and transport their personal health information. Both devices offer password protection and encryption features. Analysis of the devices shows that they store their data in a Microsoft Access database. Due to a flaw in the encryption of this database, recovering the user's password can be accomplished with minimal effort. Our analysis also showed that, rather than encrypting health information with the password chosen by the user, the devices stored the user's password as a string in the database and then encrypted that database with a common password set by the manufacturer. This is another serious vulnerability. This article describes the weaknesses we discovered, outlines three critical flaws with the security model used by the devices, and recommends four guidelines for improving the security of similar devices.

■ **J Am Med Inform Assoc.** 2007;14:397–399. DOI 10.1197/jamia.M2352.

Introduction

Personal health records (PHRs) have been widely studied in informatics and hold great promise for empowering patients and increasing the accuracy and completeness of health information.[1] Most PHRs in use today are presented as secure Web sites, but some PHRs are offered as portable USB (universal serial bus) keys. In a previous article, we described a serious security threat to physicians and hospitals posed by these devices.[2] In this article we review the encryption and security features of two such devices.

Background

The two devices discussed in this article are the Personal HealthKey (CapMed, Newtown, PA) and the E-HealthKEY (MedicAlert, Turlock, CA). They have much in common—both are small devices consisting of flash memory and a USB port. Each is designed to be attached to a keychain or lanyard and carried with a patient. Both devices have facilities for storing a variety of health information, and each offers password security and encryption. According to CapMed, their device is designed so that "data is encrypted and password protected, viewable only at the user's discretion,"[3] whereas MedicAlert markets their device as having "[d]ata encryption and password protection [that] lets the user decide which information to share."[4] This article analyzes the security of these encryption and password protection schemes.

Methods

We identified manufacturers of USB-based personal health record devices and contacted them to request samples or initiate a purchase. A number of manufacturers we contacted were still in the development phase and were unable to supply us a device. We ultimately received three devices: the CapMed Personal Health Key, the MedicAlert E-HealthKEY, and the Med-InfoChip (Med-InfoChip LLC, Boynton Beach, FL). The CapMed and MedicAlert devices have encryption features, but the Med-InfoChip does not appear to offer encryption or password protection of any kind, so it is not considered in this analysis. We enabled encryption and security on both of the devices we tested, and proceeded to manually analyze the file and database structures of the devices.

Results

During the course of the analysis, we determined that although the user interface and many of the features of the devices differed, the underlying technology and database appeared to be largely the same between the two products. In fact, a CapMed press release indicates that CapMed partnered with MedicAlert to develop and market the E-HealthKEY.[5]

Both devices use a password-protected software application to view the personal health record. Each device has an emergency function that allows responders to access a subset of medical information without a password. But each required a password, set by the user, to access the full personal health record.

Further analysis of the devices revealed that both store their data in a Microsoft Access (Microsoft Corporation, Redmond, WA) database file, accessed through the Microsoft Jet database engine. This file was password pro-

Affiliations of the authors: Department of Medical Informatics and Clinical Epidemiology, Oregon Health & Science University (AW, DFS), Department of Medical Informatics, Northwest Permanente, PC (DFS), Portland, OR.

Correspondence and reprints: Adam Wright, Department of Medical Informatics and Clinical Epidemiology, Oregon Health & Science University, 3181 Sam Jackson Park Rd., Portland, OR 97239; e-mail: <wrightad@ohsu.edu>.

Received for review: 12/15/2006; accepted for publication: 4/13/2007.

tected, so it could not be opened directly in Microsoft Access without the password. The password used to lock this database is not the same as the password the user inputs to access the PHR.

The file format chosen for the database has a significant weakness, known at least as far back as August 1998. The password needed to view such a database is stored directly in the database file, at a fixed offset, scrambled against a fixed string with a logical operator.[6] To break the encryption, we extracted the string from the file and reversed the logical operator. This instantaneously yielded the password used to encrypt the database. We then used this password to open the database in Access, where we could view its contents.

Our analysis revealed that instead of encrypting their contents with the password chosen by the user, the devices instead store the user's password as a string in the database, and then encrypt that database with a common password fixed by the manufacturer, which was the same across both devices.

Using more strict formalism, if we define sk to be an encryption function with key k, PHRdata to be the health data to be protected, and pw to be the user's password, we would expect the encryption function to be:

$$\text{encodedData} = s_{pw}(\text{PHRdata})$$

However, the two devices discussed here encode their data according to:

$$\text{encodedData} = s_{c}(\{\text{PHRdata, pw}\})$$

where c is a constant key known to the manufacturer and is consistent across devices.

Insofar as the constant key c is known or can be discovered, the security of both PHRdata and pw can be breached. Even if the manufacturers keep c secure and it cannot be determined cryptanalytically, this encryption scheme allows the manufacturer to decrypt the devices. Although this has the potential benefit of allowing the manufacturer to restore access to the device should the user forget his or her password, the user may or may not want the device manufacturer to be able to access the contents of the secure device should it enter into their possession.

Discussion

Fundamentally, the encryption scheme these two devices use has three major weaknesses:

1. Each device encrypts the database according to a common key instead of using the password chosen by the user. Once this key has been compromised on one device, all devices are potentially vulnerable.
2. Even if the key is not compromised, because it is common across devices and known to the manufacturer, the manufacturer has the ability to view the personal health data stored on any of its devices if it regains physical possession of the device.
3. The manufacturer relies on the security of a third-party database engine with well-known vulnerabilities.

The database engine used in this application, Microsoft Access, is widely used, and we suspect that there are other clinical systems that may have similar vulnerabilities. Encryption weaknesses such as this one are, however, preventable.

One question that may be important to consider is whether encryption is necessary at all. Some patients may value the fact that their medical record, carried with them, could be accessed in an emergency, or may simply perceive no need for privacy. Our concern is that by offering password protection and encryption, the devices may give the user a sense of security that is not justified by the strength of the protections. Also, even if good encryption is available, users may unintentionally dilute the protection by choosing weak passwords, so user education should go along with any attempts to harden the devices.

Recommendations

We recommend four guidelines for successfully using encryption in such applications:

1. Encrypt data according to the key provided by the user, and never to a common key.
2. Avoid encryption approaches that allow the manufacturer to decrypt data, unless the user grants explicit permission for the manufacturer to have access.
3. Do not rely on the security of third-party encryption or security schemes unless you can verify them to a degree of certainty. Several well-known cryptographic algorithms, such as AES,[7] Blowfish,[8] and TwoFish,[9] have been widely studied, and their security properties are fairly well understood. Free, open-source encryption libraries such as OpenSSL (http://www.openssl.org/) also are available.
4. Include an encryption expert on any project for which data security is important. Many programmers do not have a thorough understanding of encryption, so it is important to choose an expert with a fundamental knowledge of the theory of encryption. The best practice is to use an independent outside security auditor to review all encryption plans and implementations.

The consumer empowerment working group of the American Health Information Community has recommended a certification process for personal health records. It is worth considering whether such a certification process should include requirements relating to encryption, and possibly even an inspection process.

Conclusion

Encryption is a useful tool, and when implemented correctly can provide high security to an application. Encryption is doubly important in cases such as PHRs or any mobile application for which physical access to the device containing the data store is relatively unrestricted. With proper attention to security principles and careful implementation, we believe it is possible to create personal health record systems that are secure and provide patients with the appropriate level of confidentiality.

References ■

1. Tang PC, Ash JS, Bates DW, Overhage JM, Sands DZ. Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption. J Am Med Inform Assoc 2006;13:121–6.
2. Wright A, Sittig DF. Security threat posed by USB-based personal health records. Ann Intern Med 2007;146:314–5.

3. CapMed. CapMed Introduces Personal HealthKey Secure, Private, Portable Medical Records Enabled By Flash-Based Technology From M-Systems' DiskOnKey. 2003. Available at: http://www.capmed.com/news/press/press_release_virt_20.asp. Accessed August 10, 2006.
4. Foundation M. MedicAlert E-HealthKEY. 2006. Available at: http://www.medicalert.org/E-Health/. Accessed August 10, 2006.
5. CapMed. Bio-Imaging Technologies' CapMed Division Partners with Medicalert to Market Portable Electronic Medical Record. 2005. Available at: http://www.capmed.com/news/press/200506_medicalert.asp. Accessed August 10, 2006.
6. Microsoft Corporation. Important Aspects of Password and Encryption Protection. 2003. Available at: http://office.microsoft.com/en-us/ork2003/HA011403111033.aspx. Accessed April 9, 2007.
7. Daemen J, Rijmen V. The design of Rijndael: AES—The Advanced Encryption Standard. New York: Springer, 2002.
8. Schneier B. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). Fast Software Encryption, Cambridge Security Workshop; 1993. New York: Springer-Verlag, 1993:191–204.
9. Schneier B. The TwoFish encryption algorithm: a 128-bit block cipher. New York: Wiley, 1999.